Considering Side Effects in Service Interactions in Home Automation - an Online Approach

Michael Wilson[1], Mario Kolberg[2], and Evan H. Magill[2]

[1] Sysnet Ltd.
457 Sauchiehall Street, Glasgow G2 3LG
michael.wilson@ieee.org
[2] Department of Computing Science and Mathematics
University of Stirling, Stirling. FK9 4LA
{mko,ehm}@cs.stir.ac.uk

Abstract. The feature or service interaction problem within home networks is an established topic for the FI community. Interactions between home appliances, services and their environment have been reported previously. Indeed earlier work by the authors introduced a device centric approach which detects undesirable behaviour between appliances and their effects on the environment.
However this previous work did not address side-effects between components of the modelled environment. That is some appliances do not only affect the environment through their main function, but may do so also in other ways. For instance, an air conditioner cools the air as its main function, but also decreases the humidity as a side-effect.
Here we extend our earlier approach to handle such side effects effectively and discuss previously unreported results.

1 Introduction

The intelligent home is a reality. Increasingly, newly built homes are equipped with intelligent home equipment for improved audio visual experiences, security and comfort. These devices are controlled by software services provided by a number of vendors.

Typically, these services are run from a central point within the home, a residential gateway which may take the form of a set top box. The OSGi (Open Services Gateway Initiative) gateway [1] is one such gateway which is able to run and manage services. Devices will register with the gateway, allowing services to use a multitude of devices. OSGi becomes the *glue* which connects devices and services [2]. However, with such interworking between different services and devices, unexpected or undesirable outcomes are inevitable.

Unexpected or undesirable outcomes are known as Service Interactions, or Feature Interactions [3]. This is when the action of one service, or device, has an impact on another. The phenomenon is well understood and documented within the telephony domain as it has been the focus of research for over a decade with work widely published. However, the problem has also been identified and investigated in other domains, such as E-mail [4] and web-services [5], and more recently, within home automation [6–9].

1.1 Service Interactions in the Home

Although the service interaction problem in the home is similar to the feature interaction problem in telephony, there are some differences. The main difference is that many more interactions happen indirectly. They happen through an additional level the environment. Here, the environment can be room temperature or movement in the room, for example.

Previous approaches in telephony do not use the environment to detect interactions. Offline, a-priori and captive environment approaches are not suitable because in the home a service can behave differently depending on the devices available and how a service is configured. Further, the configuration of services and devices in the home can easily change as home networking protocols have been designed to specifically support ad hoc networking (UPnP for example).

Services such Home Ventilation and Air Conditioning, Security, and Power Control Services are able to use a number of devices to achieve their goals. However, since services are automatically controlling several devices, each carrying out their own role, it is inevitable some conflicts will occur.

This section highlights the issue of negative interactions. Feature interactions can occur for a number of reasons. The most common are conflicting goals and broken assumptions [3]. Services pursue specific goals, e.g. a Power Control service switches off unnecessary appliances to save energy. However, a feature of a security service might act to switch on lights to pretend the home is occupied. Clearly, the goals of the two services conflict. Similarly, services need to make certain assumptions about their environment. For instance, one assumption of the security service is that when activated, nobody is at home, therefore appliances will not be used and there should be no movement. However, the climate control service may control the window blinds to prevent the sun from unnecessarily heating up the home. Here the assumption of the security service that no appliances will be used, is violated by the climate control service.

The remainder of the paper is organised as follows. Section 2 discusses the employed runtime approach together with its constituting parts: the three layer model, environmental variables, service priorities, device database and the service interaction manager. This section introduces the novel concept of side effects in feature interaction handling. It also contains a short description of the dynamic behaviour of the approach. Section 3 presents the working of the approach using a detailed example. Section 4 discusses results together with a decsription of the testbed, service examples and interaction scenarios. At the end of that section the results of the approach are compared against the taxonomy for service interactions in home networks by Kolberg et al [6]. This is followed by some concluding remarks.

2 A Runtime Environmental Approach

Traditional approaches to feature interaction have been service centric, concentrating on the actions, goals, and assumptions of the services. Further, many of these approaches are off-line. Work by Nakamura et al [8] and Metzger and

Webel [9] does concentrate on the device and environment, rather than just the service. However, these approaches are off-line. Although off-line approaches are useful for detecting some interactions, they only work in a system where all services and devices are known. In the home this is unlikely as services from many vendors will work together. Even if all services were known, the configuration of the devices will be unclear as devices will join and leave the network.

Also, the services which control devices may behave differently depending on the devices available at runtime. Indeed the services may change over time. This makes an off-line approach unworkable within the home. Therefore, in this paper a runtime, feature manager based approach is advocated. While feature manager based approaches require central control, this is not a problem in the home as typically the residential gateway is centralised and all devices and services register on the same platform. The approach has to be transparent to users. If a negative interaction occurs, it should be avoided. However, if the interaction is positive, it should be allowed, as this is desirable.

2.1 Basic operation of the Feature Manager

Some interactions can be detected at the device level - two services try to use one device. However, others seem unconnected and cannot be verified by monitoring network messages. For instance there is an interaction between the security service and the climate control service where the climate control service operates a fan while the security service is active. The movement of the fan triggers the security service. This conflict does not happen at the device level, it happens elsewhere in the environment. The movement of the fan is detected by the sensors of the security service. For this reason, the environment layer is included and is central to this approach. The approach works by controlling access to the environment layer by using access locking.

The approach assumes that there will be a residential gateway in the home where services are managed and executed [10]. The services which run on the gateway will send commands directly to the devices. Since these messages are sent at runtime, a live manager is required. This manager must intercept messages which are sent from the service to the device and determine whether a particular command will cause a negative interaction. Crucially, the manager distinguishes between positive and negative interactions. Positive interactions result from two or more services or devices working together towards a common goal. In contrast, a negative interaction is where the outcome is undesirable.

After analysing an instruction sent to the device, if the manager decides the action will cause either no interaction or a positive interaction, the message will be forwarded to the device. However, if the manager detects a negative interaction, the message will not be allowed to proceed to the device. Instead, the manager adds an entry to its log and discards the message.

2.2 The three layer approach

The advocated approach uses three layers. The top layer is the service layer which contains the services that automate the home. These services may use one

or a combination of home appliances (devices) which are located in the second layer. The approach has been designed in a way that the underlying device communication protocol is not relevant making the approach very flexible.

The device layer can contain two types of devices: input devices and output devices. An input device, such as a thermometer, will only monitor an aspect of the environment (e.g. room temperature) and return readings to services. An output device, such as a heater, will alter the environment in some way (increase temperature). Where a physical device contains both functions (input and output) it would be seen as two devices in the approach similarly to the UPnP approach which splits devices in such a way.

Finally, the bottom layer is the environmental layer containing environmental variables. These variables are a representation of a rooms environment. Examples of environment variables include: room movement, room temperature, room lighting levels, humidity, and smoke.

2.3 Locking devices and variables

Within the device and environment layers, controlling access to devices and environmental variables is achieved through locking. For devices, locking is only necessary for output devices – as input devices do not affect their environment in any way.

Simply locking a variable is too crude for this approach. For example, when a heater is active the room temperature variable would be locked. Since the variable is locked, another heater would not be able to also heat the room. Although there is an interaction here, it is a positive one and should be allowed. On the other hand it would not be correct to share the temperature variable between a heater and an air-conditioner as the two devices have conflicting goals. Concepts were used from the Biased Protocol [11]. Especially, the concepts of Shared Locks and Exclusive Locks have been adapted in this approach.

One important aspect with home devices are side effects. Quite often a device does not only affect the environment through its primary function, but also in other ways. It is not sufficient to treat side effects in the same way as primary effects. This approach decrees that each device can affect the environment through one primary effect and zero or more side effects. If a device affects the environment in two equally important ways, the device should be split into two separate logical devices. For instance, a television affects the light levels and the noise variables. Both can be argued to be primary effects. Hence the television needs to be split into an audio device and a video device, the first affecting the noise variable and the second the light variable as primary effects.

If a service wishes to lock a device, or a device wants to lock a variable, they must be locked with one of five options:

- *NS : Not Shared.* The variable or device is locked and may not be altered by any other device or service. This lock is similar to the exclusive locks in the biased protocols.

– *S+ : Shared, but increase only.* The variable is shared on the condition that anyone wishing to use the variable must increase it. Therefore, two devices may lock a variable with S+ if they both increase value. This allows two heaters to operate.
– *S- : Shared, but decrease only.* Like the previous setting, the variable is shared on the condition that anyone wishing to alter the variable must decrease it.
– *S± : Shared.* The variable or device is shared and it is unknown whether the variable will be decreased or increased in value. This lock is not compatible with S- or S+ because S± could go either way (increase or decrease). This lock type can be likened to the shared lock type from the biased protocol.
– *SE : Side Effect.* This flag is placed on an environment variable by a device. Like the locks, SE must be placed on a variable before the device is allowed to run. A device can place the SE flag on any environment variable, provided the variable is not already locked with NS. If a variable is locked with S+, S- or S±, SE can be added to the variable.

By using these locks, devices are able to cooperate and work to a common goal while conflicting actions will be avoided. Table 1 summarises the list of locks above and shows the combination of locks which are allowed ($\surd$).

	NS	S+	S-	S±	SE
NS					
S+		√			
S-			√		
S±				√	
SE		√	√	√	√

Fig. 1. Locking – allowed combinations.

Many services or devices may lock a device or variable with matching S+, S or S±. However, only one service or one device has access when a device or variable is set with NS.

Once a lock has been set, it is sometimes not clear when the task is complete and the lock can be lifted. This approach assumes that a session begins when a service starts using a device and finishes when the service closes or switches the device off. Therefore, when a lock is placed on a device, the lock is valid until the service finishes using the device.

2.4 Service Priorities

By using the locks as described above, a strict first come – first served order is implied. However, in certain circumstances this is not adequate. For instance suppose the home is being burgled and the VCR is in use by the entertainment service to record the owners favourite show (locked with NS). Since the VCR is in use, the home security service is unable to access the device to record the intruder. Service priorities have been introduced to resolve such scenarios. Each service gets a priority number assigned. A priority value may change as new services are added to the home, or a users preferences change. The priorities

of services will range from 1 to n, where n is the total number of services in the gateway. Priority 1 is the lowest and n is the highest. There is one special priority: -1 (no priority) for services that have not been assigned a priority.

2.5 The Device Database

In order to understand the effects of the actions of devices on the environment the approach needs to have knowledge on what each device does. For instance if a heater is turned on, it will produce heat, which in turn affects its environment and increases the room temperature.

Therefore a database which holds device details is required. The database of devices is likely to be constantly growing as new devices come onto the market. There is an issue with the location of such a database. A local database would be rather large and difficult to mange and keep up to date with new devices. In contrast, if the database were to be hosted remotely, all homes would be able to share the same data but a constant connection to the internet is required. Also in the future appliances may store information about their behaviour internally, however, this cannot be assumed at present. Consequently, for the purpose of this paper, a remote database is used.

When querying the remote database, the manager supplies a device type, such as heater, air-fan, and television. The type supplied is matched with the description within the database and the device specifications in an XML format are returned. An Example is shown in Figure 2. This shows a heater is an output device, and the default usage for a service is NS. The XML also shows the environmental variables this device affects, Temperature and Humidity. Clearly, increasing the temperature is the primary goal for using a heater, hence the change in temperature is the primary effect for the heater. The change in humidity levels is a side effect as influencing humidity is usually not the primary goal of using a heater. The default value for Temperature is S+ as temperature will be increased. For Humidity, the value is SE, to indicate the side effect. Various other attributes are included in the device description, such as the default lock values for the device. This makes it possible for this approach to work without having any input from the service fully operational on its own.

2.6 The Service Interaction Manager (SIM)

The manager operates as a standard service within the service gateway within the home. The manager intercepts messages and authorises them after checks have been carried out. There are two possible outcomes: either forward the message to the device or reject the message. To make the decision the manager analyses the state of the required devices and associated environmental variables based on an internal image of the state of all devices.

The manager generates the view of the home by searching the gateway for all device objects registered and consulting the remote device database obtaining all associated environmental variables required for each device. If a device joins or leaves the network this change will be picked up by the manager which will

```
1. <Device>
2. <DeviceType>Heater</DeviceType>
3. <DeviceIO>Output</DeviceIO>
4. <DefaultDeviceUsage>NS</DefaultDeviceUsage>
5. <Action>
6.    <Name arg="on">deviceOn</Name>
7.    <SuggestedDeviceUsage use="NS" />
8.    <EnvironmentalVariable name="Temperature"
9.        defaultValue="S+" duration="3" />
10.   <EnvironmentalVariable name="Humidity"
11.       defaultValue="SE" duration="3" />
12. </Action>
13. <Action>
14.    <Name arg="off">deviceOff</Name>
15.    <SuggestedDeviceUsage use="" />
16.    <EnvironmentalVariable name="Temperature"
17.        defaultValue="" duration="0" />
18.    <EnvironmentalVariable name="Humidity"
19.        defaultValue="" duration="0" />
20. </Action>
21. </Device>
```

Fig. 2. Description of Device Type 'Heater'

update its view. If the manager authorises a command to a device, the manager records the new device state, thus keeping itself up to date and consistent.

If a device is controlled directly by the user, perhaps the user has pressed play or stop buttons on the VCR, this new state must be recorded in the managers view. Importantly, even if the SIM notes that the new state causes an interaction, the new state still has to be recorded as this *is* the current state of the device.

2.7 Dynamic Behaviour of the approach

A model which shows the home services as well as devices and their relationship with the environmental variables has been developed (Fig. 3). All services, devices and environmental variables would normally be included. However, for clarity, only one service, two devices (an input device and output device) and one environmental variable has been included here.

In this example, the service layer contains one service and the service name is shown (Figure 3(b)), along with the priority of the service (a). In the device layer Sensor T is surrounded by a double rectangle whereas Device Y has a single rectangle. The double rectangle represents an input device and the single rectangle represents an output device. Both primary and side effects are shown.

The fact this is an output device is shown by the direction of the arrows (f) and (g). The environmental layer shows the variable Z that is monitored by the sensor device. On the other hand, the arrow (g) shows that the output device will also affect the environment variable Z.

Figure 3(c) shows the lock for controlling access to the device itself. This lock is set by the service. Generally, a service will use NS, as it is unlikely to

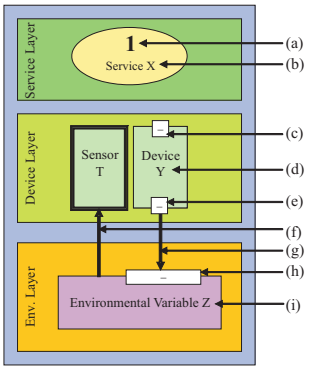

Fig. 3. Outline of Model

want another service using the device while it is in use. If a service does not understand how to set the lock for the device, a default lock from the device description database, which is normally NS, is used. Once access to the device has been gained, the manager must ensure the device is able to gain access to all required environmental variables (i). The variables a device affects when it carries out specific actions are obtained from the remote device database. Figure 3(e) shows the proposed lock for the environmental variable. The arrow (g), points to the variable where the lock is to be set. (h) shows the current lock of the environmental variable. Locks are shown both at the bottom of devices and on the environmental variables because the variable may already be locked by another device.

3 A running Example – Interaction between Climate Control and Security Services

The security service monitors movement in the home through the use of PIR sensors. If there is movement detected an alarm is raised by ringing a bell. The climate control service keeps the house at a comfortable temperature by using air conditioning, heating, and by opening windows. Clearly, while the alarm is armed, it would not be appropriate for the climate control to open windows in the home. Not only would opening the windows cause movement, triggering the alarm, but the two services have conflicting goals. The security service aims to keep the home secure, while the climate control is to cool the home by opening the window which renders the home insecure.

The climate control service uses a thermometer (input device) to get the room temperature. In this home, there is also an external thermometer, so the outside temperature can be obtained, but for clarity this is not shown in the figures. To control the temperature the service can use a heater to increase temperature, an air-conditioner to decrease temperature and a window. The open window will change room temperature either up or down, depending on the outside

temperature. During opening and closing, the window will create movement as a side effect. Similarly, the heater and air condition will affect humidity as a side effect. Therefore, among the three devices, three environmental variables are required: temperature, movement, and humidity.

The security service requires a motion sensor to detect movement, and two output devices: an alarm control panel, which is used to set or disable the alarm, and an alarm bell. When the alarm is triggered, the bell device is used to draw attention. These devices affect movement and sound as environmental variables.

It is assumed that the service priorities have been set by the user (or a service provider on their behalf). The climate control has been set with the lowest priority, 1, and the security service with 2. Therefore, the security service has priority over the climate control service. This setup is shown in Figure 4.

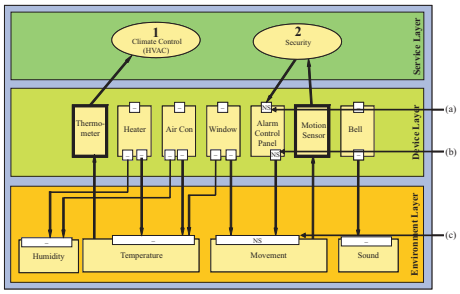

Fig. 4. Arming the security service

As the security service is armed, the security service accesses the alarm device. Since this device is not in use, it is able to gain access and lock it with NS as it does not want anyone else using the device (Figure 4(a)). Using the default values from the device description database, the manager knows that for the arm command, the movement variable should be locked using NS (Figure 4(b)). Since the variable is not currently locked, this value is set in the variable (Figure 4(c)).

The security service is now armed and monitoring the home. The climate control service is also active, but it does not require any devices, other than the thermometer notifying the service of a change in room temperature.

3.1 Handling the Interaction

Assume that the temperature within the home starts to rise and the climate control service (knowing that it is cooler outside) is to open a window to allow the cool air in. If the service interaction manager were not in place, the climate control service would open the window making the home insecure and also triggering the alarm.

With an active manager the command to open the window will be checked by the manager. First, the climate control service must be able to access the window. Since the window is unused, the service is granted access and sets the device lock with NS, Figure 5(b). The dashed lines indicate temporary links prior to authorisation by the manager.

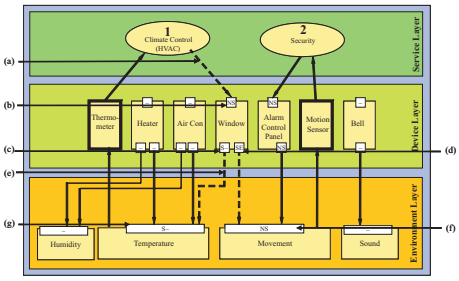

Fig. 5. Avoiding the interaction

The manager can now attempt the devices proposed locks for the environmental variables. The two proposed lock boxes are shown (Figure 5(c) and (d)). Since it has been determined that the temperature outside is colder than inside, the device needs to lock the temperature variable with S- (c) and (g). Also, as opening the window will cause movement as a side effect, the device needs to lock movement with SE (d). However, this fails as movement is already locked with NS (f), and NS and SE are not compatible (cf. Figure 1). Furthermore, the lock on the movement variable can not be overwritten as it has been set by a service with a higher priority. Hence the window device is unable to open and the interaction has been successfully avoided.

If the climate control service truly was a smart service, it would search for an alternative way of cooling the room and select the air condition. This would be successful and both services would work in harmony.

4 Experimentation and Results

To show the effectiveness of the approach, experimentation on 11 scenarios was carried out. In the following the testbed is introduced followed by a summary of the obtained results.

4.1 Testbed

The testbed (cf. Fig. 6 includes a selection of devices (UPnP and X.10), the service management framework (OSGi) and the home control services. X.10 was chosen as one of the control protocols because it is currently widely used in homes. The protocol is popular because of the availability [12] and cost of the components. Also, typical household devices (e.g. lamps, fans, heaters) can be used and no new wiring is required as it uses the power lines as the transport medium. For experimentation, a range of X.10 modules were used including a CM11 gateway model, lamp modules (together with lamp), appliance modules (with a desk fan) and motion sensors as well as a virtual X.10 window opener.

UPnP is a new home networking protocol and is growing with more OEM companies becoming members. However, at present, only UPnP routers and

internet gateways are available to buy off the shelf. Devices required for this test-bed, such as heaters or air conditioners are not available yet. Therefore, virtual devices were used.

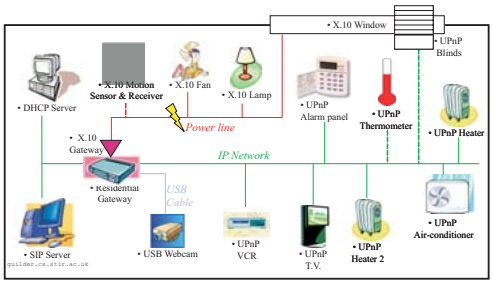

Fig. 6. The testbed used for experimentation

Using virtual devices offers flexibility for both creating and controlling the device. To create a virtual UPnP device, a UPnP SDK, including UPnP stack, was used. For this experiment, an open source UPnP stack was used CyberLink [13]. As well as forming the base for the UPnP devices, it was also used to create the UPnP driver for the service management framework. Each UPnP device developed had a simple GUI for user input (e.g. set device on or off, or set channel, etc.). These are the kind of controls one would expect on simpler devices. The XML device description and service definitions followed those published by the UPnP Forum. Where definitions for devices were not available, the basic device specification was used [14]. The following UPnP devices were created and used in the testbed:

Thermometer – a device which reads room temperature. When queried it returns the room temperature. Also when the temperature changes, subscribed parties are notified. As this is not a real device, a slider-bar is used to manually change the temperature.

Heater - a simple heater device with two options: on or off. Since this is a virtual device and to mirror what would happen in reality, the heater finds all thermometers in the room and increases their temperature. This addition did not change the functioning of the device.

Air-conditioner - like the heater, this is a simple device which has two options: on or off. Similar to the heater, to simulate reality the device finds thermometers and decreases the temperature.

Television - a device which tunes into a channel and displays the picture sent by a TV station. The TV has a number of functions: on and off, change the channel, and volume up and down. For experimental purposes, a TV station had to be created. This is a Java server which devices can connect to. Images are then sent by the server at regular intervals (1 per second).

VCR - the VCR records to file any images it was sent. The user can then tune the TV into the VCR and play back the recorded images. The options available in this device are: on and off, play, record and set channel to record.

Window blinds - This device simulates a small motor which can open and close the blinds accordingly.

Additionally, two support devices were used in the test-bed. These were: a USB web camera, and a SIP Server [15]. The IBM Service Management Framework was used as the implementation of the OSGi gateway. An X.10 and UPnP driver had to be developed as these are not included in the SMF. The UPnP driver is based on the Cybergarage UPnP stack [13] and follows the UPnP driver specification in the OSGi standard [16]. Similarly, the X.10 driver was developed using an existing open source Java X.10 API [17]. The Java Communications API [18] was used to access the communications port. The following services were implemented on the OSGi gateway.

Heating, Ventilation and Ai-conditioning (HVAC) - This service monitors the temperature in the home and keeps it at a comfortable temperature as defined by the user.

Home Security Service (HSS) - This service is used in conjunction with an alarm device. The service can be set, and configured, through the alarm device. The role of this service is to monitor the home for intruders and alert the owner of such an event. This service also has the away from home feature which makes the home look occupied when the home is empty.

Power Control Service (PCS) - The aim of this service is to reduce the amount of power a home consumes. It does this by turning off devices when no one is home. The service can be set to turn on devices which consume a lot of energy when electricity is cheaper a washing machine, for example.

Home Entertainment Service (HES) - This service controls entertainment devices, such as the TV, stereo, and VCR/DVD devices. One of the main features of this service is that it can in principle monitor viewing habits and automatically record certain television shows. The service does allow an option to manually set the time and date to record a television programme.

Communications Support Service (CSS) - This service supports the use of email and telephone. The user can be notified of email arriving through their television. Also, any incoming telephone calls can be displayed on the screen; the user can decide whether to accept the call or not.

Humidity Control Service (HCS) – Like the HVAC service, this service controls the humidity by employing a humidifier and a dehumidifier. A humidistat is used as the input device.

Carbon Monoxide Safety Service – This service monitors the CO content in the air. If the levels are too high it will open a window and alert the user.

4.2 The test cases

For testing all of the above services were used. The security service and carbon monoxide safety service have priorities, 2 and 3 respectively, set. Through experimentation, it was found that a 'first come, first served' approach was adequate for all but safety services. However, it must be noted that this does depend on user preferences. Therefore, a user may wish to set all services to have a priority.

The following list provides details on the individual interaction scenarios.

Scenario 1: HSS:AFH vs PCS – During the absence of the home owner, the security service's away from home feature turns appliances on to give the impression that someone is at home. However, the power control service turns appliances off to save energy.
Assuming the security service is running first, it has locked the TV and lamp with *NS*, therefore no other service can access them. Consequently, when the power control service tries to gain access to turn the devices off, its request is denied. If, however, the power control service accesses and locks the devices first, since the security service has a higher priority, the power control service will be overridden by the security service. Therefore the security service will be allowed access to both the lamp and the TV.

Scenario 2: HSS:Alarm vs HES – The security service is triggered while the home entertainment service is recording a TV program on the VCR (locked *NS*). The HSS wants to record pictures from the security camera on the VCR. Because the HSS has a higher priority than the HES, the the HES has to give up control of the VCR. The VCR can then be reassigned to the security service. Thus, the picture of the burglar can be recorded on tape.
If the HSS is active first and the HES tries to record a programme then HSS:Alarm feature has acquired the VCR and locked it with *NS*. The home entertainment service is then refused as it has a lower priority.

Scenario 3: HSS:AFH vs HVAC – Here, the away from home feature of the security service is turning on lights and closing the blinds following a pattern to make the home appear occupied. However, the HVAC service wants to increase the room temperature by opening the blinds. However, the HVAC service cannot get access to the blinds as they are locked (*NS*) and it has a lower priority as the security service. The blinds remain closed.

Scenario 4: HVAC vs HSS:Alarm – The HSS service monitors movement in the home. The HVAC service is set to cool the home. Realising it is cooler outside, the climate control wants to open the window to allow cool air in creating movement and triggering the alarm service. This interaction was discussed in detail in the previous section.

Scenario 5: Within HVAC – Issue 1 – The problem here is that the climate control service has been configured incorrectly and can potentially allow both the air conditioner and heater to operate simultaneously. Since the heater increases room temperature, the temperature variable is locked with S+. The climate control then tries to turn on the air conditioner. Since the air conditioner device is not in use it gains access, but it cannot lock the room temperature variable with S– as this is incompatible with S+. Both devices have also a side effect on humidity (SE). Both devices in this scenario can successfully lock the humidity variable with SE. Hence for the interaction detection this is without consequence.

Scenario 6: Within HVAC – Issue 2 – This interaction is similar to the interaction above, but here the heater and air conditioner are not used at the same time, but sequentially in a loop, that is the home is heated up, only to be cooled down again, and then be heated again and so on.

Unfortunately, this is one type of interaction which this approach cannot detect. This is because after the heater, or air conditioner, has completed its task, it releases locks on all its variables. These variables are then free to be locked by any other device.

Scenario 7: within HSS – This interaction occurs when the alarm feature is monitoring the home for intruders and the away from home feature wants to lower the blinds creating movement which in turn triggers the alarm.

In this example, the alarm is armed and the movement variable is locked with NS. When the away from home feature tries to lower the blinds, it needs access to the blind device (ok) and the movement and temperature variables. However, locking the movement variable is not permitted. Therefore, the blinds cannot be lowered and the interaction is avoided.

Scenario 8: within HVAC – The first seven scenarios have shown how the approach avoids negative interactions. However, the approach must also allow devices to cooperate and work together to achieve a common goal.

This example shows two heaters working together to heat a room quickly. The service locates two heaters. Since no other service is using the devices, access is granted. Then the temperature variable needs to be locked with S+ by both heaters. The first heater is able to lock with this value successfully. When the second heater tries to lock with S+, the variable is already in use. However, this is allowed because its value of S+ is compatible with the locked value of S+. Furthermore, because of the side effect of both heaters in terms of decreasing the humidity, the humidity variable needs to be locked by both using SE. Again this is permitted. Both heaters can operate.

Scenario 9: HES and HVAC – This is a second positive interaction. The air condition is switched on by the HVAC service and the owner is watching a movie through the entertainment service. Now the entertainment service tries to close the blinds to prevent any glares on the TV.

The air condition has locked the temperature variable with S–. The entertainment service gets access to the blind device (NS) and movement variable (SE). The temperature variable is in use (S-) but the HES service also uses S- for access. This is allowed and the blinds can close.

Scenario 10: CMSS vs HSS – The security service is armed, but as the carbon monoxide levels exceed safe levels, the CMSS tries to open a window. This scenario requires priorities. HSS is assigned a priority of 2 and the CMSS has been set priority 3. If while the security service is active, the carbon levels exceed a safe limit, the CMSS must open a window. Access to the window is granted (NS), however, access to the movement variable is locked by the HSS service (NS). But since CMSS has a higher priority than HSS, CMSS is able to get access to the variable. The window can be opened.

A similar interaction may occur between the climate control service and CMSS. The climate control service may be trying to heat the home (using heaters). By opening a window this would let cold air in. The conflict is the same – the CMSS wants to open the window, whereas the other services (HVAC or HSS) want to remain shut.

Scenario 11: HCS and HVAC – This scenario illustrates side effect of the heater on the humidity variable (SE). Clearly, there is an issue between the humidity control service and the HVAC service on the humidity variable. However, as the heater accesses the humidity variable as a side effect (SE) it is compatible with the lock by the HCS (S+), and both services are allowed to work simultaneously.

4.3 Results

Table 1 summarises the results obtained from experimentation. All positive interactions have been allowed by the approach, and all negative interactions except for the looping ones have been avoided. Hence the approach can handle successfully a very large proportion of interactions in the home.

Scenario	*pos./neg.*	*Description of Interaction*	*handled by approach*
1	negative	HSS:AFH with PCS	√
2	negative	HSS:Alarm with HES	√
3	negative	HSS:AFH with HVAC	√
4	negative	HVAC with HSS:Alarm	√
5	negative	Within HVAC (issue 1): wasting energy	√
6	negative	Within HVAC (issue 2): Looping	×
7	negative	Within HSS AFH with Alarm features	√
8	positive	Within HVAC	√
9	positive	HES with HVAC	√
10	negative	CMSS with HSS	√
11	negative	HCS with HVAC	√

Table 1. Summary of results

Table 2 shows the types of interactions which the approach avoids. Kolberg et al. [6] identified four types of interaction, with looping a special case of SAI. The approach is not able to handle the looping interaction type because the locked variables are released, and can then be locked by another device.

Interaction type	*Handled by approach*
Multiple action interaction (MAI)	√
Sequential action interaction (SAI)	√
Looping (Special case of SAI)	×
Shared trigger interaction (STI)	√
Missed trigger interaction (MTI)	√

Table 2. Interaction types handled by the approach

5 Conclusions

In this paper we presented a *runtime* technique which prevents negative interactions while allowing positive interactions to occur. The approach considers incompatible accesses to devices and aspects of the environment (environmental variables). This paper introduces side effects of devices on the environment as a novel concept in feature interaction improving the accuracy of the approach.

With side effects, minor influences of devices on the environment (a heater reduces humidity) can be captured in the model.

A comprehensive case study with a wide variety of services and devices is presented demonstrating the power of the approach. It is shown that except for looping interactions, the approach can handle all other types of interactions identified in home networks. Because the approach builds on the OSGi gateway, it is protocol independent. That is, there are no restrictions put on which communication protocols can be used by the devices.

Interaction between services in the home are a real issue. Services form a number of sources will "meet" for the first time when deployed into the home. However, it cannot be expected that a technical expert will be at hand to sort out any incompatibilities between services and devices. It just has to work.

References

1. OSGi: The Open Services Gateway Initiative. `http://www.osgi.org`.
2. P. Dobrev, D. Famolari, C. Kurzke, and B.A. Miller. Device and service discovery in home networks with OSGi. *IEEE Communications Magazine*, 40(8), August 2002.
3. E. J. Cameron, N. Griffeth, Y.-J. Lin, M. E. Nilson, W. Shnure, and H. Velthuijsen. Towards a Feature Interaction Benchmark for IN and Beyond. *IEEE Communications Magazine*, 31(3):64–69, March 1993.
4. R. Hall. Feature interactions in electronic mail. In *[19]*, pages 67–82, May 2000.
5. M. Weiss. Feature interactions in web services. In *[20]*, pages 149–156, June 2003.
6. M. Kolberg, E. Magill, and M. Wilson. Compatibility issues between services supporting networked appliances. *IEEE Communications Magazine*, 41(11), November 2003.
7. M. Wilson, E.H. Magill, and M. Kolberg. An online approach for the service interaction problem in home networks. In *IEEE Consumer Communications and Networking Conference (CCNC-2005), Las Vegas, USA.*, January 2005.
8. M. Nakamura, H. Igaki, and K. Matsumoto. Feature interactions in integrated services of networked home appliances. In *[21]*, pages 236–251, June 2005.
9. A. Metzger and C. Webel. Feature interaction detection in building control systems by means of a formal product model. In *[20]*, pages 105–122, June 2003.
10. F. T. H. den Hartog, M. Balm, C. M. de Jong, and J. J. B. Kwaaitaal. Convergence of residential gateway technology. *IEEE Communications Magazine*, 42(5), May 2004.
11. A. Silberschatz and P.B. Galvin. *Operating System Concepts.* Addison Wesley, 5th edition, 1998.
12. LetsAutomate.co.uk. `http://www.letsautomate.co.uk/`, viewed: 18/05/2007.
13. CyberLink development package for UPnP devices. `http://www.cybergarage.org/net/upnp/java/index.html` viewed: 18/05/07.
14. UPnP Basic Device Specification. `http://www.upnp.org/standardizeddcps/basic.asp` viewed: 18/05/07.
15. SIP Express Router (SER). `http://www.iptel.org/ser/` viewed: 18/05/07.
16. The Open Services Gateway Initiative. *OSGi Service Platform, Release 3.* IOS Press, 2003.
17. Jesse Peterson X.10 API. `http://www.jpeterson.com/category/software/`, viewed: 18/05/2007.
18. Java Communications API. `http://java.sun.com/products-/javacomm/index.jsp` viewed: 18/05/07.
19. M. Calder and E. Magill, editors. *Feature Interactions in Telecommunications and Software Systems VI.* IOS Press (Amsterdam), May 2000.
20. D. Amyot and L. Logrippo, editors. *Feature Interactions in Telecommunications and Software Systems VII.* IOS Press (Amsterdam), June 2003.
21. S. Reiff-Marganiec and M. Ryan, editors. *Eigth International Conference on Feature Interactions in Telecommunications and Software Systems.* IOS Press, June 2005.